



Objetivo:

- Consolidar los conocimientos adquiridos en clase sobre Java.

Enunciado:

En un programa de ordenador, las facturas tienen necesariamente un conjunto de datos de los productos de los cuales comprar y un conjunto de datos del cliente, un total (valor decimal) y una fecha de la factura.

Los datos del cliente son la cadena de caracteres nombre, cedula, y el entero fiabilidad de pago, mientras que los datos del detalle de la factura son sólo su producto, cantidad, valor.

Cada clase tendrá los métodos para leer y fijar (“set” y “get”) todos sus atributos.

También se debe incluir en la clase Factura un método de “Borrado”, que no devuelve ni recibe ningún parámetro.

Los productos que venden la empresa son de carácter de computadores y tiene un id, descripción, precio unitario, stock, iva (si se calcula el iva o no).

Los atributos serán privados y tendrán métodos públicos para acceder a ellos. Se pide dibujar el diagrama UML de las clases Factura, Datos_del_cliente y Datos_del_producto y generar el sistema en java para realizar los procesos de CRUD para generar facturas, productos, clientes.

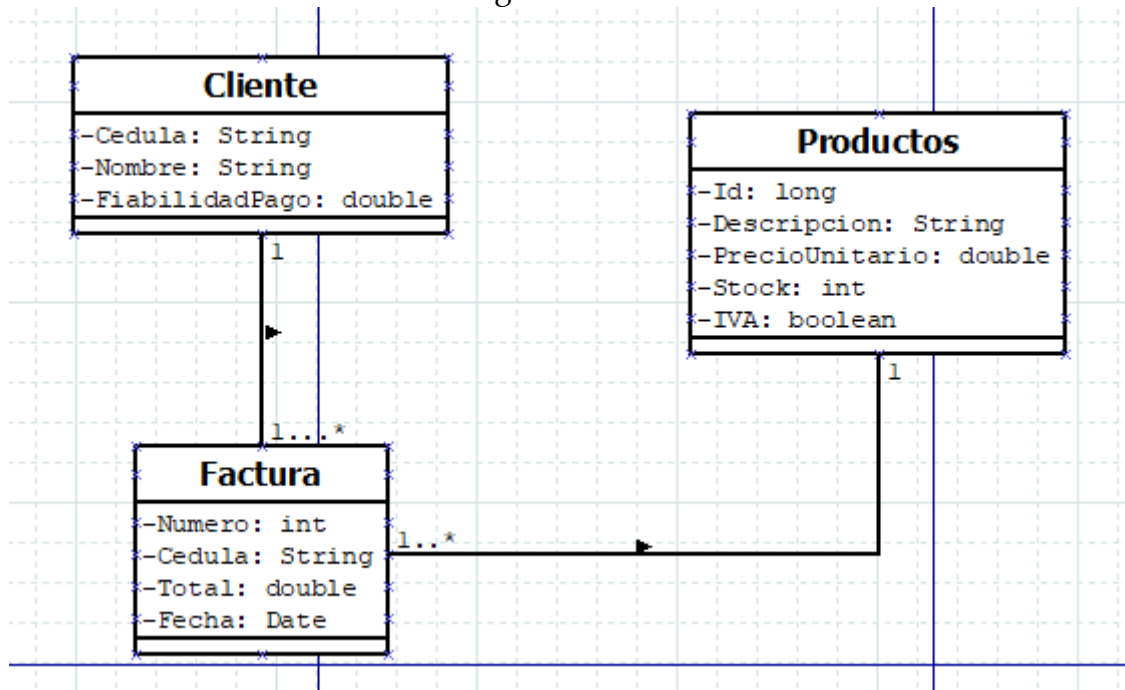
Se calificará el avance con los siguientes criterios de evaluación:

- Diagrama de Clases: 25%
- MVC: 25%
- Usabilidad – Vista - Menús: 25%
- Ejecución, pruebas y sustentación 25%

Entrega: Subir al Git el documento en formato PDF de la practica y código hasta las **23:55** del domingo **06 de Junio del 2021**.



Diagrama del Dia





Examen

06/03/2021

Para este programa se va aplicar el MVC (Modelo Vista Controlador).

Comenzamos por el Modelo. El modelo es donde toda la lógica y datos, se pueden relacionar con la base de datos y se usa el encapsulamiento.

Se crea un paquete con el nombre Modelo, donde van a ir todas las creaciones de las variables, los constructores, los gets y sets, y el toString, de las respectivas secciones que se tiene que crear desde el diagrama que se realizó en el día (Cliente, Productos, Factura).



```
package modelo;

/**
 *
 * @author Usuario
 */
public class Cliente {
    private String Nombre;
    private String Cedula;
    private double fiabilidadPago;

    public Cliente(String Nombre, String Cedula, double fiabilidadPago) {
        this.Nombre = Nombre;
        this.Cedula = Cedula;
        this.fiabilidadPago = fiabilidadPago;
    }

    public String getNombre() {
        return Nombre;
    }

    public void setNombre(String Nombre) {
        this.Nombre = Nombre;
    }

    public String getCedula() {
        return Cedula;
    }

    public void setCedula(String Cedula) {
        this.Cedula = Cedula;
    }

    public double getFiabilidadPago() {
        return fiabilidadPago;
    }

    public void setFiabilidadPago(double fiabilidadPago) {
        this.fiabilidadPago = fiabilidadPago;
    }

    @Override
    public String toString() {
        return "Cliente{" + "Nombre=" + Nombre + ", Cedula=" + Cedula + ", fiabilidadPago=" + fiabilidadPago + '}';
    }
}
```



Productos

```
package modelo;

public class Producto {
    private long id;
    private String Nombre;
    private String descripcion;
    private double precioUnitario;
    private int Stock;
    private boolean Iva;

    public Producto(long id, String Nombre, String descripcion, double precioUnitario, int Stock, boolean Iva) {
        this.id = id;
        this.Nombre=Nombre;
        this.descripcion = descripcion;
        this.precioUnitario = precioUnitario;
        this.Stock = Stock;
        this.Iva = Iva;
    }

    public long getId() {
        return id;
    }

    public void setId(long id) {
        this.id = id;
    }

    public String getNombre() {
        return Nombre;
    }

    public void setNombre(String Nombre) {
        this.Nombre = Nombre;
    }

    public String getDescripcion() {
        return descripcion;
    }

    public void setDescripcion(String descripcion) {
        this.descripcion = descripcion;
    }

    public double getPrecioUnitario() {
        return precioUnitario;
    }

    public void setPrecioUnitario(double precioUnitario) {
        this.precioUnitario = precioUnitario;
    }

    public int getStock() {
        return Stock;
    }

    public void setStock(int Stock) {
        this.Stock = Stock;
    }

    public boolean getIva() {
        return Iva;
    }

    public void setIva(boolean Iva) {
        this.Iva = Iva;
    }

    @Override
    public String toString() {
        return "Producto{" + "id=" + id + ", Nombre=" + Nombre + ", descripcion=" + descripcion + ", precioU"
    }
}
```



Examen

06/03/2021

Factura

```
package modelo;

import java.util.Date;
import modelo.Cliente;

/**
 *
 * @author Usuario
 */
public class Factura {
    private int Numero;
    private int Cantidad;
    private double Valor;
    private String fecha;
    private Cliente cliente;
    private Producto producto;

    public Factura(int Numero, Cliente cliente, Producto producto, int Cantidad, double Valor, String Fecha) {
        this.Numero=Numero;
        this.cliente = cliente;
        this.producto=producto;
        this.Cantidad = Cantidad;
        this.Valor = Valor;
        this.fecha = fecha;
    }

    public Factura(int Numero, int Cantidad, double Valor, Date Fecha, Cliente cliente, Producto producto) {
        this.Numero = Numero;
        this.Cantidad = Cantidad;
        this.Valor = Valor;

        this.cliente = cliente;
        this.producto=producto;
    }

    public int getNumero() {
        return Numero;
    }

    public void setNumero(int Numero) {
        this.Numero = Numero;
    }
}
```



Examen

06/03/2021

```
public int getCantidad() {  
    return Cantidad;  
}  
  
public void setCantidad(int Cantidad) {  
    this.Cantidad = Cantidad;  
}  
  
public double getValor() {  
    return Valor;  
}  
  
public void setValor(double Valor) {  
    this.Valor = Valor;  
}  
  
public String getFecha() {  
    return fecha;  
}  
  
public void setFecha(String fecha) {  
    this.fecha = fecha;  
}  
  
public Cliente getCliente() {  
    return cliente;  
}  
  
public void setCliente(Cliente cliente) {  
    this.cliente = cliente;  
}  
  
public Producto getProducto() {  
    return producto;  
}  
  
public void setProducto(Producto producto) {  
    this.producto = producto;  
}  
  
@Override  
public String toString() {  
    return "Factura{" + "Numero=" + Numero + ", Cantidad=" + Cantidad + ", Valor=" + Valor + ", cliente=" + cliente + '}';  
}  
}
```



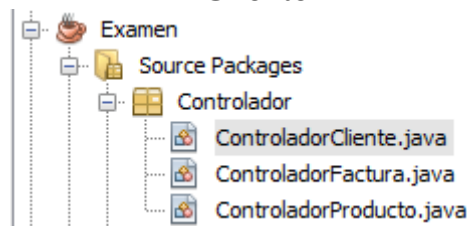
Examen

06/03/2021

Después del Modelo son los Controladores, un Controlador se encarga de relacionar el modelo con la vista realizando distintas funciones.

Después de la creación del modelo, se crea un paquete llamado Controlador donde se aplicara el CRUD (Created, Read, Update, Delete), irán los métodos, importaciones, las distintas clases del CRUD ya mencionado y los gets y setters, para poder realizar el controlador de las respectivas secciones (Cliente, Productos, Factura).

Cliente





Examen

06/03/2021

```
package Controlador;

import java.util.ArrayList;
import java.util.List;
import modelo.Cliente;
import modelo.Producto;

/**
 *
 * @author Usuario
 */
public class ControladorCliente {
    private List<Cliente> listaCliente;
    private Cliente seleccionado;

    public ControladorCliente() {
        listaCliente=new ArrayList<>();
    }

    public boolean crear(String Nombre, String Cedula, double fiabilidadPago){
        return listaCliente.add(new Cliente(Nombre, Cedula, fiabilidadPago));
    }

    public Cliente buscar(String Cedula){
        for(Cliente cliente:listaCliente){
            if(cliente.getCedula().equals(Cedula)){
                seleccionado=cliente;
                return cliente;
            }
        }
        return null;
    }

    public boolean actualizar(String Nombre, String Cedula, double fiabilidadPago){
        Cliente cliente = buscar(Cedula);
        if (cliente!=null){
            int posicion = listaCliente.indexOf(cliente);
            cliente.setFiabilidadPago(fiabilidadPago);
            cliente.setNombre(Nombre);
            listaCliente.set(posicion, cliente);
            return true;
        }
        return false;
    }
}
```




Examen

06/03/2021

```
public boolean eliminar(String Cedula){
    Cliente cliente=buscar(Cedula);
    return listaCliente.remove(cliente);
}

public List<Cliente> getListaCliente() {
    return listaCliente;
}

public void setListaCliente(List<Cliente> listaCliente) {
    this.listaCliente = listaCliente;
}

public Cliente getSeleccionado() {
    return seleccionado;
}

public void setSeleccionado(Cliente seleccionado) {
    this.seleccionado = seleccionado;
}

}
```



Examen

06/03/2021

Factura

En factura se hace un auto incremental con una clase llama generar Número, un auto incremental nos ayuda para controlar que no se repitan los números.

```
package Controlador;

import java.util.ArrayList;
import java.util.Date;
import java.util.List;
import modelo.Factura;
import modelo.Cliente;
import modelo.Producto;

/**
 *
 * @author Usuario
 */
public class ControladorFactura {
    private List<Factura> listaFactura;
    private Factura seleccionado;

    public ControladorFactura() {
        listaFactura=new ArrayList<>();
    }

    public int generarNumero(){
        return(listaFactura.size()>0)? listaFactura.get(listaFactura.size()-1).getNumero()+1:1;
    }

    public boolean crear(int Numero, Cliente cliente , Producto producto, int Cantidad, double Valor, String Fecha){
        return listaFactura.add(new Factura(generarNumero(), cliente, producto, Cantidad, Valor, Fecha));
    }

    public Factura buscar(int Numero){
        for(Factura factura:listaFactura){
            if(factura.getNumero()==(Numero)){
                seleccionado=factura;
                return factura;
            }
        }
        return null;
    }

    public boolean actualizar(int Numero, Producto producto, int Cantidad, double Valor){
        Factura factura = buscar(Numero);
        if (factura!=null){
            int posicion = listaFactura.indexOf(factura);

            factura.setCantidad(Cantidad);
            factura.setValor(Valor);
            listaFactura.set(posicion, factura);
            return true;
        }
        return false;
    }
}
```



Examen

06/03/2021

```
public boolean eliminar(int Numero){
    Factura factura=buscar(Numero);
    return listaFactura.remove(factura);
}

public List<Factura> getListFactura() {
    return listaFactura;
}

public void setListaFactura(List<Factura> listaFactura) {
    this.listaFactura = listaFactura;
}

public Factura getSeleccionado() {
    return seleccionado;
}

public void setSeleccionado(Factura seleccionado) {
    this.seleccionado = seleccionado;
}
```

Producto

Igualmente, en producto se hace un auto incremental para controlar que no se repitan los números.

```
package Controlador;

import java.util.ArrayList;
import java.util.List;
import modelo.Producto;

/**
 *
 * @author Usuario
 */
public class ControladorProducto {
    private List<Producto> listaProducto;
    private Producto seleccionado;

    public ControladorProducto(){
        listaProducto=new ArrayList<>();
    }

    public long generarId(){
        return(listaProducto.size()>0)? listaProducto.get(listaProducto.size()-1).getId()+1:1;
    }

    public boolean crear(String Nombre, String descripcion, double precioUnitario, int Stock, boolean Iva){
        return listaProducto.add(new Producto(generarId(),Nombre, descripcion, precioUnitario, Stock, Iva));
    }

    public Producto buscar(String Nombre){
        for(Producto producto:listaProducto){
            if(producto.getNombre().equals(Nombre)){
                seleccionado=producto;
                return producto;
            }
        }
        return null;
    }

    public boolean actualizar(String Nombre, String descripcion, double precioUnitario, int Stock){
        Producto producto = buscar(Nombre);
        if (producto!=null){
            int posicion = listaProducto.indexOf(producto);
            producto.setDescripcion(descripcion);
            producto.setPrecioUnitario(precioUnitario);
            producto.setStock(Stock);
            listaProducto.set(posicion, producto);
            return true;
        }
        return false;
    }
}
```



Examen

06/03/2021

```
public boolean eliminar(String Nombre){
    Producto producto=buscar(Nombre);
    return listaProducto.remove(producto);
}

public List<Producto> getListProducto() {
    return listaProducto;
}

public void setListaProducto(List<Producto> listaProducto) {
    this.listaProducto = listaProducto;
}

public Producto getSeleccionado() {
    return seleccionado;
}

public void setSeleccionado(Producto seleccionado) {
    this.seleccionado = seleccionado;
}
```

Después del controlador viene la vista, la vista se encarga en mostrar todas las opciones, operaciones, etc, al cliente o sujeto.

Después se crea otro paquete con el nombre Vista para mandar a mostrar a cada uno de las secciones.

Cliente





Examen

06/03/2021

```
package vista;

import Controlador.ControladorCliente;
import java.util.Scanner;
import modelo.Cliente;

/**
 * @author Usuario
 */
public class vistaCliente {
    public Scanner teclado;
    private ControladorCliente controladorCliente;
    public vistaCliente(ControladorCliente controladorCliente){
        teclado=new Scanner(System.in);
        this.controladorCliente=controladorCliente;
    }

    public void menu() {
        int opcion=0;
        do{
            System.out.println("1. Crear \n 2. Actualizar \n 3. Buscar \n 4. Eliminar \n 5. Listar \n 6. Sair");
            opcion=teclado.nextInt();
            switch(opcion){
                case 1:this.crear(); break;
                case 2:this.actualizar(); break;
                case 3:this.buscar(); break;
                case 4:this.eliminar(); break;
                case 5:this.listar(); break;
            }
        }while(true);
    }
    public void crear()
    {
        System.out.println("Ingreso: \n Nombre");
        String Nombre=teclado.next();
        System.out.println("Cedula");
        String cedula=teclado.next();
        System.out.println("Fiabilidad de Pago");
        double fiabilidad=teclado.nextDouble();
        System.out.println("Res: "+ controladorCliente.crear(Nombre,cedula, fiabilidad));
    }
    public void actualizar(){
        System.out.println("Ingreso: \n Nombre");
        String Nombre=teclado.next();
        System.out.println("Ingreso: \n Cedula");
        String Cedula=teclado.next();
        System.out.println("Fiabilidad de Pago");
        double fiabilidad=teclado.nextDouble();
        System.out.println("Res: "+controladorCliente.actualizar(Nombre, Cedula, fiabilidad));
    }
    public void buscar() {
        System.out.println("Ingreso: \n Cedula");
        String cedula=teclado.next();
        System.out.println(controladorCliente.buscar(cedula));
    }
    public void eliminar(){
        this.buscar();
        System.out.println("Res: "+controladorCliente.eliminar(controladorCliente.getSeleccionado().getCedula()));
    }
    public void listar(){
        for(Cliente cliente : controladorCliente.getListCliente()){
            System.out.println(cliente);
        }
    }
}
```



Examen

06/03/2021

Factura

```
package vista;

import Controlador.ControladorFactura;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Scanner;
import modelo.Factura;
import Controlador.ControladorCliente;
import Controlador.ControladorProducto;
import modelo.Cliente;
import modelo.Producto;

/**
 *
 * @author Usuario
 */
public class vistaFactura {
    public Scanner teclado;
    private ControladorFactura controladorFactura;

    public vistaFactura(ControladorFactura controladorFactura){
        teclado=new Scanner(System.in);
        this.controladorFactura=controladorFactura;
    }

    public void menu(){
        int opcion=0;
        do{
            System.out.println("1. Crear \n 2. Actualizar \n 3. Buscar \n 4. Eliminar \n 5. Listar \n 6. Sair");
            opcion=teclado.nextInt();
            switch(opcion){
                case 1:this.crear(); break;
                case 2:this.actualizar(); break;
                case 3:this.buscar(); break;
                case 4:this.eliminar(); break;
                case 5:this.listar(); break;
            }
        }while(true);
    }
}
```



Examen

06/03/2021

```
public void crear() {
    System.out.println("Ingreso: \n Numero de Cedula");
    String cedula=teclado.next();
    Cliente cliente;
    ControladorCliente controlador = null;
    cliente = controlador.buscar(cedula);
    System.out.println("Producto");
    String nombreProducto=teclado.next();
    Producto producto;
    ControladorProducto productoControlador = null;
    producto=productoControlador.buscar(nombreProducto);
    System.out.println("Cantidad");
    int cantidad=teclado.nextInt();
    System.out.println("Valor");
    double valor=teclado.nextDouble();
    System.out.println("Fecha");
    String fecha=teclado.next();

    System.out.println("Res: "+ controladorFactura.crear(0,cliente, producto,cantidad,valor, fecha));
}
public void actualizar(){
    System.out.println("Ingreso: \n Producto");
    String nombreProducto=teclado.next();
    Producto producto;
    ControladorProducto productoControlador = null;
    producto=productoControlador.buscar(nombreProducto);
    System.out.println("Cantidad");
    int cantidad=teclado.nextInt();
    System.out.println("Valor");
    double valor=teclado.nextDouble();

    System.out.println("Res: "+controladorFactura.actualizar(0, producto,cantidad,valor));
}
public void buscar() {
    System.out.println("Ingreso: \n Numero");
    int numero=teclado.nextInt();
    System.out.println(controladorFactura.buscar(numero));
}
public void eliminar(){
    this.buscar();
    System.out.println("Res: "+controladorFactura.eliminar(controladorFactura.getSeleccionado().getNumero()));
}
    public void listar(){
        for(Factura factura : controladorFactura.getListFactura()){
            System.out.println(factura);
        }
    }
}
```



Examen

06/03/2021

Producto

En producto en la parte se realiza un if para saber si el producto contiene IVA o no, el tipo de la variable IVA es tipo boolean.

```
package vista;

import Controlador.ControladorProducto;
import java.util.Scanner;
import modelo.Producto;

/**
 *
 * @author Usuario
 */
public class vistaProducto {
    public Scanner teclado;
    private ControladorProducto controladorProducto;

    public vistaProducto(ControladorProducto controladorProducto){
        teclado=new Scanner(System.in);
        this.controladorProducto=controladorProducto;
    }

    public void menu(){
        int opcion=0;
        do{
            System.out.println("1. Crear \n 2. Actualizar \n 3. Buscar \n 4. Eliminar \n 5. Listar \n 6. Sair");
            opcion=teclado.nextInt();
            switch(opcion){
                case 1:this.crear(); break;
                case 2:this.actualizar(); break;
                case 3:this.buscar(); break;
                case 4:this.eliminar(); break;
                case 5:this.listar(); break;
            }
        }while(true);
    }
}
```




Examen

06/03/2021

```
public void crear() {
    System.out.println("Ingrese: \n Nombre del producto");
    String Nombre=teclado.next();
    System.out.println("Descripcion del producto");
    String descripcion=teclado.next();
    System.out.println("PrecioUnitario");
    double preUnitario=teclado.nextDouble();
    System.out.println("Stock");
    int stock=teclado.nextInt();
    System.err.println("Contiene Iva Si/No: ");
    boolean iva;
    String contieneIva=teclado.next();
    if(contieneIva=="Si"){
        iva=true;
    } else {
        iva=false;
    }
    System.out.println("Res: "+controladorProducto.crear(Nombre, descripcion, preUnitario, stock, iva));
}

public void actualizar() {
    System.out.println("Ingrese: \n Nombre");
    String Nombre=teclado.next();
    System.out.println("Descripcion");
    String descripcion=teclado.next();
    System.out.println("Precio Unitario");
    double PrecioUni=teclado.nextDouble();
    System.out.println("Stock");
    int stock=teclado.nextInt();
    System.out.println("Res: "+controladorProducto.actualizar(Nombre, descripcion, PrecioUni, stock));
}

public void buscar() {
    System.out.println("Ingrese: \n Nombre");
    String nombre=teclado.next();
    System.out.println(controladorProducto.buscar(nombre));
}

public void eliminar() {
    this.buscar();
    System.out.println("Res: "+controladorProducto.eliminar(controladorProducto.getSeleccionado().getNombre()));
}

public void listar() {
    for(Producto prodcuto : controladorProducto.getListaProducto()){
        System.out.println(prodcuto);
    }
}
}
```